Types of manual Testing

1.Black Box Testing

This form of testing is also known as behavioral testing. It aims to analyze the application's behavior from the perspective of the end-user. This method is not concerned with details of the internal system, or how it processes specific actions at the code level, but only tests requirement compliance.

Black box testing can be further divided into functional and non-functional testing.

Functional Testing: This is a form of black box testing where we seek to confirm that an application works according to the requirements. Three of the several types available are as follows:

Smoke Testing: This testing is executed once the build has been deployed. Testers perform a smoke test on basic functionality to ensure that the application is stable for further testing. For example: if the login test fails, we cannot test the application further.

Regression Testing: Whenever a build is deployed in the testing environment, we perform regression testing. This type of testing's main objective is to make sure that any new changes are working as expected and do not affect the previous or existing modules.

User Acceptance Testing: This form of testing focuses on functionality and usability. It is usually the final assessment before the product is ready for release.

2. Non-Functional Testing:

In this type of black box testing, we evaluate the application's performance and speed under certain circumstances. Below are two of the most often used types:

Usability Testing: The application's ease of use for the end-user or customer is tested with this method.

Localization Testing: Localization testing aims to ensure that the application has been customized for the targeted global audience regarding language and culture.

3.**Exploratory Testing**

In this kind of manual testing, the tester can learn about the application, so design test cases accordingly. This testing is based on the tester's critical thinking, where real-time scenarios are created for a higher level of quality. Exploratory testing can be beneficial when requirements have not been sufficiently elaborated. Hence, the tester explores the application by learning the functionality. Real-time thinking is useful for finding loopholes in a minimum time frame. The aim is to emphasize the tester's ability to explore the application and broaden understanding.

4.**Integration Testing**

This type of testing is used for evaluating how the application behaves when integrated with multiple modules. The purpose of integration testing is to verify the alignment of interfaces among the different modules integrated as a whole.

Integration testing includes using the following two popular approaches:

Bottom-Up Approach: In this approach, we move steadily from the bottom module to the top module. This activity progresses until we integrate all the modules and test them completed as a whole.

Top-Down Approach: This approach is opposite to the above. Testing starts from the top module (which is tested as a separate unit). It gradually moves to the bottom by integrating the modules one by one.

5.**System Testing**

This testing method is performed to evaluate the complete, integrated system according to the specified requirements. With system testing, the emphasis is to compare the particular program with its original objectives. It is a destructive technique because it checks for potential failure points in the specifications and design, and also makes sure that goals are met, including customer satisfaction.

Testing can be performed on objectives, documentation, and the program itself. System testing compares all parts for compatibility with everything else. However, it should be noted that it is one of the more difficult techniques to perform because there is no specific methodology that can guide system testing.

System testing uses the following techniques:

Install ability Testing: This method is used to ensure that the application installs correctly on the respective systems and environments, as mentioned in the guidelines document.

Recovery Testing: Testers need to confirm that the application can recover gracefully from unfavorable circumstances.

6.GUI or Graphical User Interface Testing

The most important element for the end-user is the GUI of any application. The well-known phrase, “the first impression is the last impression,” runs true for the GUI. Because it’s the entry point for how the user interacts with the application, it’s where opinions will be formed. GUI testing aims to verify that the graphical components are appropriately aligned and work according to the specifications for various environments and devices. These include buttons, checkboxes, text fields, dropdowns, and colors, etc.

A GUI benefits from two types of testing as follows.

UI Testing: The look and feel of the application play a pivotal role in any application’s success. UI testing’s primary focus in relation to GUI testing is to ensure that all the fonts, buttons, checkboxes, tabs, menu, images, etc. are suitably aligned and in the right position.

Functionality Testing: The purpose of functionality testing with GUI testing is to verify the respective functionalities’ navigation in both positive and negative scenarios.

7.User Acceptance Testing (UAT)

Once testing within the organization is complete, the product is released to the client to test for acceptability. This part of testing occurs in a production-like environment where the client, or a limited number of end-users, uses the system and reports any discrepancies found.

Acceptance testing has the following types.

Alpha Testing: Alpha testing is acceptance testing executed within the organization that developed the application. Selected people within the organization are given access to use and evaluate the application and provide feedback to inform further improvements.

Beta Testing: Beta testing is the final phase where the application is released to the public, although usually to a limited number of users at first. Organizations then gather feedback from those users and check if there is anything that needs to be improved.